Predicting Bank Loan Eligibility

G. MallikarjunaRao[1], Sk. Mohammed Asif[2], A. Sairam Naik[3], V. Rakesh[4], Sd. Rizwana[5]

[1,2,3,4]Student Dept, of CSE, Narasaraopeta Engineering College, Narasaraopeta, Guntur, (D.T) AP,India

[5]Asst.Professor, Dept, of CSE, Narasaraopeta Engineering College,Narasaraopeta,Guntur,(D.T) AP,India

Mallikarjuna14273@gmail.com[1], shaikasifsv@gmail.com[2], Sairamnaik121@gmail.com[3], Vengalasettyr74@gmail.com[4], rizwana.nec@gmail.com[5]

ABSTRACT

Processes across various industries, including real estate, security, biology, and the financial sector, are being revolutionized by machine learning algorithms. One of the most laborious responsibilities in the banking sector is the process. The speed, effectiveness, and accuracy of loan approval processes can all be enhanced by contemporary technology, such as machine learning models. In order to forecast loan eligibility, this study provides six machine learning techniques (Random Forest, Gradient Boost, Decision Tree, Support Vector Machine, K-Nearest Neighbor, and Logistic Regression). The historical dataset "Loan Eligibility Dataset," which is accessible on Kaggle and licensed under Database Contents License (DBCL) v1.0, was used to train the models. On Kaggle's Jupiter Notebook cloud platform, Python programming libraries were used to process and analyses the dataset. The Random Forest algorithm provided the highest performance accuracy in our study.

KEYWORDS

Effective ML algorithms, KNN, SVM,and loan approval prediction.

INTRODUCTION

For financial institutions, determining whether aborrower will be eligible for a bank loan is a crucial step in determining their creditworthiness. In the past, banks have evaluated loan applications using manual processes,which can be labor-intensive and prone to mistakes.Large datasets can be analyzed by machine learning to find patterns and trends that can be used to forecast the likelihood of loan approval, which has the potential to automate and enhance the loan approval process.The goal of this project is to create a predictive model based on machine learning algorithms that can reliably forecast loan eligibility based on a variety of variables, including income, credit score, employment history, and debt-to-income ratio. Banks and other financial institutions can use the model toreduce the risk of loan default by deciding whether to approve loan applications.

REVIEW OF RELATED LITERATURE

Machine learning-based eligibility prediction for bank loans has become a hot topic in recent years. Many studies have been done in this field with the goal of increasing the precision of forecasts about loan eligibility. Arora et al(2019)
.'s work created a loan eligibility prediction model utilising several machine learning algorithms on a dataset of 500 loan applications. With an accuracy rate of 85.2%, they discovered that the Random Forest method performed better than other algorithms. The important determinants of loan eligibility, such as credit score, income, and employment history, were identified by their model. Another work by Lohiya et al. (2020) used a dataset of 1,000 loan applications to create a Support Vector Machine model for predicting loan eligibility (SVC).

METHODOLOGY

The process of predicting bank loan eligibility using machine learning entails developing a predictive model that can correctly categorise loan applicants as either eligible or ineligible depending on particular requirements. A loan eligibility prediction model can be created using the following methodology:

1. Data Gathering: Gather the pertinent information that will help determine a loan applicant's eligibility. This information can contain financial and personal details like age, earnings, credit score, employment status, loan amount, loan term, etc.

2. Data Pre-processing: Prepare the data beforehand to deal with missing values, outliers, and inconsistent data. Data transformation, feature engineering, and data cleaning may be necessary for this.

3. Divide the data into training and testing sets. In order to train the model, the training set will be used, and the testing.

DATASET

The process of predicting bank loan eligibility using machine learning entails developing a predictive model that can correctly categories loan applicants as either eligible or ineligible depending on requirements. A loan eligibility prediction model can be created using the following methodology:

Data Gathering: Gather the pertinent information that will help determine a loan applicant's eligibility. This information can contain financial and personal details like age, earnings, credit score, employment status, loan amount, loan term, etc.

Data Pre-processing: Prepare the data beforehand to deal with missing values, outliers, and inconsistent data. Data transformation, feature engineering, and data cleaning may be necessary for this.

Divide the data into training and testing sets. In order to train the model, the training set will be used, and the testing.

To build a machine learning model to predict bank loan eligibility, you'll need a dataset containing relevant features and corresponding labels. Here are some potential features you might consider including in your dataset:

The labels in your dataset will be whether or not an applicant was granted a loan. You can use historical data from the bank tocreate your dataset, labelling each application as either approved or denied based on the bank's decision.

After you get your dataset, you can pre-process it to address potential problems like missing values and categorical variables. The data can then be divided into training and testing sets, and a model can be trained on the training data using a machine learning technique like logistic regression or a decision tree.Lastly, you may assess the performance of your model using measures like accuracy, precision, recall, and F1 score on the testing data.

Dataset description

Variable Name Details Data Type	Variable Name Details Data Type	Variable Name Details Data Type
Loan ID Loan identification number (Unique I.D.)	n ID Loan identification number (Unique I.D.)	Loan ID Loan identification number (Unique I.D.)
Numeric \sGender	Numeric \sGender	Numeric \sGender
candidate's gender	candidate's gender	candidate's gender
Categorical \sMarried	Categorical \sMarried	Categorical \sMarried
marital status of the applicant	marital status of the applicant	marital status of the applicant
Categorical \sDependents	gorical \sDependents	Categorical \sDependents
Family size in numbers	mily size in numbers	Family size in numbers
Numeric \sEducation	ic \sEducation	Numeric \sEducation
applicant's educational background (Graduate or not graduate)	applicant's educational background (Graduate or not graduate)	applicant's educational background (Graduate or not graduate)
Categorical \sSelf-employed	l \sSelf-employed	Categorical \sSelf-employed
Employment status of the applicant (self-employed: yes; employed/others: no)	Employment status of the applicant (self-employed: yes; employed/others: no)	Employment status of the applicant (self-employed: yes; employed/others: no)
Income for Categorical Applicant	Income for Categorical Applicant	Income for Categorical Applicant
Salary/income per month of the applicant	Salary/income per month of the applicant	Salary/income per month of the applicant

DATA PRE-PROCESSING ANALYSIS

Machine learning for predicting bank loan eligibility requires the pretreatment and analysis of data. The primary steps in data pretreatment and analysis are listed below.
Data cleaning remove duplicates, missing values, outliers, and unrelated variables from the data.

Data exploration and visualization Investigate the data by running statistical computations and visualizing the results to learn more about the connections between the variables and the goal variable (loan approval). This action entails:
Calculate descriptive statistics for the variables in the dataset, including mean, median, mode, standard deviation, and correlation coefficient.

To illustrate the distribution of the variables and spot any patterns or trends that might be present in the data, create charts, histograms, and scatterplots. Choose the characteristics that are most important to you.

Use one-hot encoding or label encoding to transform categorical variables into numerical variables. To enhance the performance of the model, scale the variables so that the mean is zero and the standard deviation is one.To assess the model's performance on untested data, divide the data into training and testing sets.

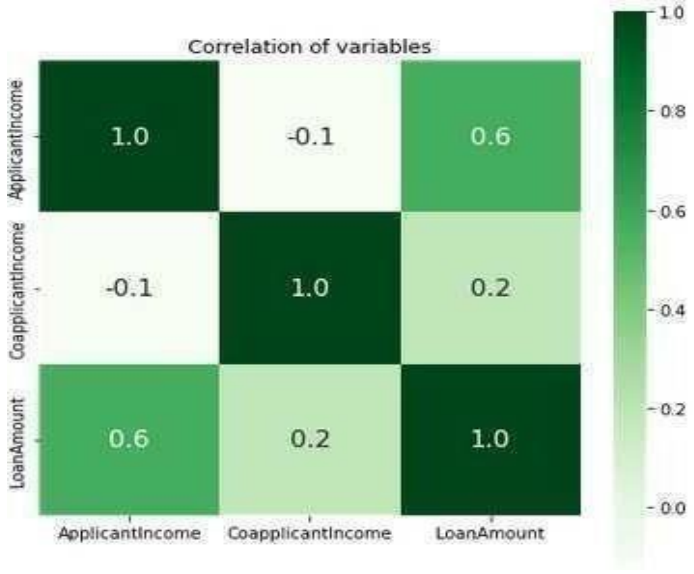
Treatment for Imbalance: Resample the data to address any imbalances, such as oversampling or under sampling, in the target variable (loan approval).Outliers Treatment: Replace extreme numbers with lessextreme ones to handle any outliers in the data using techniques like Winsorization.

Missing Values Treatment: Address any missing valuesin the data using techniques like imputation, where the missing values are replaced with estimates based on other variables in the dataset. You can enhance the quality of the data and boost the precision of the machine learning model used to forecast bank loan eligibility by carrying out these data pretreatment and analysis processes.

Synthetic Minority Oversampling Technology (SMOTE) This method works incredibly well for resolving imbalanced classification issues, a commonsource of mistake in ML models. For the aforementioned rationale, Nitesh Chawla et al. (2002) presents a novel method to produce synthetic data for oversampling purposes in their study. SMOTE generates fake data using the k-nearest neighbor technique. SMOTE begins by selecting random data Analyzing the data, we discovered that the dataset includes more married applications than unmarried ones, more male applicants than female applicants, and more applicants with good credit than those with low credit (0).

Also, the dataset's important variables are associated in Fig. 1 below, demonstrating that the attribute with the highest positive association to loan amount is applicant income.

correlation of key variables in the dataset



MODEL DEVELOPMENT AND RESULT

An ML model's performance is explained by evaluation metrics in this study, The Confusion Matrix and F1 Score were utilized as indicators of performance. The confusion matrix categorizes the number of accurate and inaccurate predictions made by an ML model. True positive situations are those that the model actually predicted to be positive. False positives are actual positive cases that the model mis predict, according to definition True negative cases are those that the model actually predicts to be negative.

The F1 score is the average of the precision and recall numbers. The following formula is provided:
A person's performance can be evaluated using their F1 score. Binary classification model, such as one that forecasts a person's loan eligibility from a bank.

The F1 score can be determined as follows assuming that we have a dataset with two classes: eligible (positive class) and not eligible (negative class), and that we have trained a binary classification model on this dataset:

Initially, we must determine the precision and recall values for the eligible positive class:

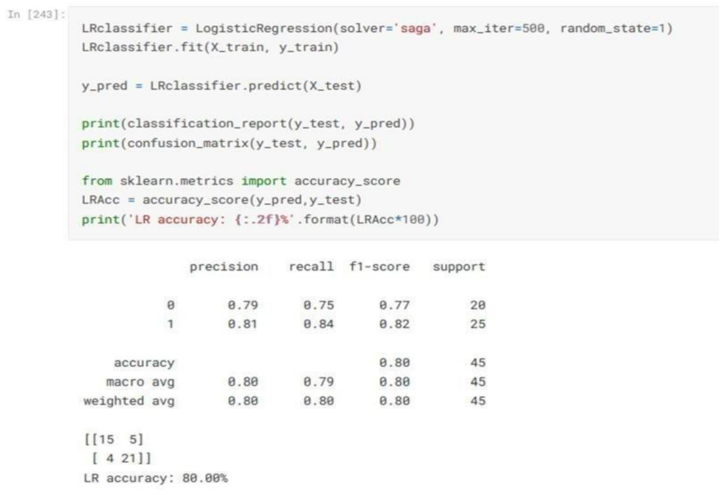
The number of cases that are accurately identified as eligible is known as the "True Positive" (TP)False positive (FP):
the proportion of cases that are projected to be eligible but are not.

False negative (FN) occurrences are those that are projected to be ineligible but are actually eligible.
Precision is equal to TP/(TP + FP). TP / (TP = recall)

LOGISTIC REGRESSION (LR) ALGORITHM

A well-liked approach for binary classification issues, such as determining whether or not a person qualifies for a bank loan, is logistic regression (LR). It is a supervised learning method that predicts the likelihood of an instance, given its attributes, falling into the positive class (qualified for a bank loan). Each feature's contribution to the final prediction is determined by a set of weights (coefficients) that the algorithm learns.Data Preparation: The dataset must first be cleaned, go through pre-processing, and be divided into training and testsets. The next step is featuring engineering, which entails choosing the pertinent features that can forecast the goal variable (bank loan eligibility) and transforming them into a form that the LR algorithm can use.Then, we fit the LR model using the training data by applying an appropriate optimization algorithm to optimize the weights (coefficients) (such as gradient descent). In order to reduce the error between the anticipated probabilities and the actual labels, the algorithm modifies the weights during training.Evaluation of the LR model: Following training, we assessthe LR model's effectiveness using the test set. Several evaluation metrics, including precision.The logistic function, which converts the input data to a number between 0 and 1, is used by the Logistic Regression (LR) technique to model the likelihood that an instance falls into the positive class (qualified for a bank loan). The logistic function's formula is as follows:Where z is the linear combination of the input features andtheir accompanying weights: z = w0 + w1x1 + w2x2 +... + wn*xn, sigmoid(z) = 1 / (1 + exp(-z))where the weights of the input features x1 to xn are,correspondingly, w1 to wn and w0 is the bias term.Using the aforementioned method, we first get the logistic function's output, or anticipated probability, which we can then use to determine whether or not a given instance qualifies for a bank loan. We can then establish a threshold(such as 0.5

Evaluation result of our LR model:



```
In [243]:
LRclassifier = LogisticRegression(solver='saga', max_iter=500, random_state=1)
LRclassifier.fit(X_train, y_train)

y_pred = LRclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
LRAcc = accuracy_score(y_pred,y_test)
print('LR accuracy: {:.2f}%'.format(LRAcc*100))
```

```
              precision    recall  f1-score   support

           0       0.79      0.75      0.77        20
           1       0.81      0.84      0.82        25

    accuracy                           0.80        45
   macro avg       0.80      0.79      0.80        45
weighted avg       0.80      0.80      0.80        45

[[15  5]
 [ 4 21]]
LR accuracy: 80.00%
```

K-NEAREST NEIGHBOR (KNN) ALGORITHM

K-Nearest Neighbor (KNN) is a straightforward but efficient algorithm for classification issues, such as determining whether or notto grant a borrower a bank loan. It is a supervised learning algorithm that categorizes new examples based on their resemblance to the preexisting instances in the training set using a distance-based method.The following are the fundamental actions to apply the KNN algorithm for predicting bank loan eligibility using machine learning:

Data Preparation: First, we must clean, pre-process, and divide the dataset into training and test sets.

Feature Engineering: Next, we must choose the pertinent features that can accurately forecast the goal variable (bank loan eligibility) and restructure them into a form that the KNN algorithm can use.

Developing the KNN model: Using the training set, we then employ Predicting Loan Eligibility: To determine whether a new instance is qualified for a bank loan or not, we measure the distances between it and each instance in the training set, and then we choose the k-nearest neighbors based on the distances that are the smallest. We can utilize a variety of distance measures, including the Manhattan distance, the Euclidean distance, and the cosine similarity. The majority class label of the new instance's k-nearest neighbors is then used to decide the class label for the new instance.

Assessing the KNN model: Using the test set, we assess the performance of the KNN model after predicting the labels of the new instances in general, KNN is an easy-to-understand method that may be used to forecast the likelihood that a person will be approved for a bank loan with a fair degree of accuracy, especially when the dataset is small and the pertinent features are properly chosen. However, if the dimensionality of the data or the imbalance in the class distribution rise, the performance of the KNN algorithm may suffer.

The K-Nearest Neighbor (KNN) algorithm for predicting bank loan eligibility classifies new instances based on their resemblance to the preexisting examples in the training set using a distance-based method. The KNN algorithm's formula is as follows.

Distance calculations: Using a distance metric such the Euclidean distance, Manhattan distance, or cosine similarity, we determine the distance between a new instance and each instance in the training set. The following formula can be used to determine how far apart two occurrences, I and j, are from one another:

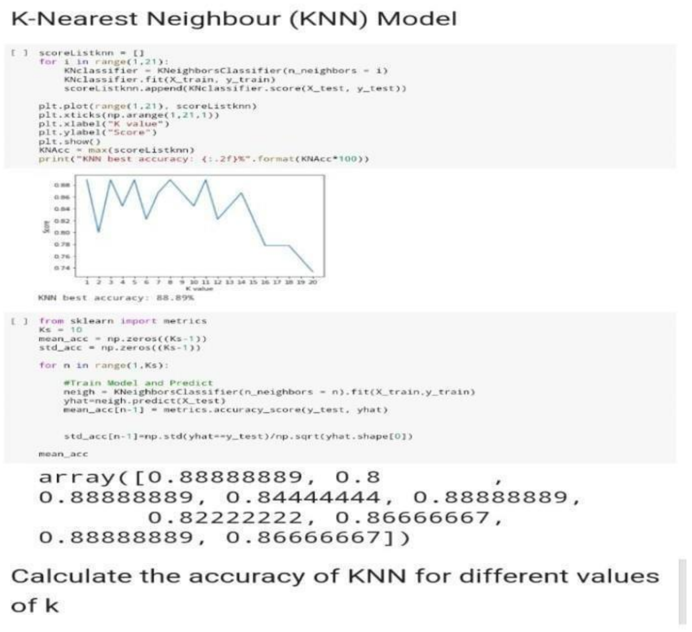
dist(i,j) is equal to sqrt(sum((xi - xj)2)).

where the feature values for instances I and j, respectively, are xi and xj.

After determining the distances, we choose the k-nearest neighbors of the new instance based on those with the least distances. Depending on the size of the dataset and the difficulty of the task, we can choose the value of k. In order to prevent ties during the majority voting stage, k is typically an odd number.

By using the majority class label of its k-nearest neighbors, we decide the class label for a new instance. The new instance is categorized as class B, for instance, if the k-nearest neighbors contain three examples of class B and two instances of class A.

The KNN algorithm, in general, is a non-parametric and lazy learning algorithm, which means it does not learn a particular model or estimate parameters from the training data. Instead, it leverages the training instances that were stored in memory to make predictions when a new instance is introduced.


K-Nearest Neighbour (KNN) Model

```
[ ] scoreListknn = []
    for i in range(1,21):
        KNclassifier = KNeighborsClassifier(n_neighbors = i)
        KNclassifier.fit(X_train, y_train)
        scoreListknn.append(KNclassifier.score(X_test, y_test))

    plt.plot(range(1,21), scoreListknn)
    plt.xticks(np.arange(1,21,1))
    plt.xlabel("K value")
    plt.ylabel("Score")
    plt.show()
    KNAcc = max(scoreListknn)
    print("KNN best accuracy: {:.2f}%".format(KNAcc*100))
```

```
KNN best accuracy: 88.89%
```

```
[ ] from sklearn import metrics
    Ks = 10
    mean_acc = np.zeros((Ks-1))
    std_acc = np.zeros((Ks-1))

    for n in range(1,Ks):

        #Train Model and Predict
        neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
        yhat=neigh.predict(X_test)
        mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)


        std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

    mean_acc
```

```
array([0.88888889, 0.8        ,
0.88888889, 0.84444444, 0.88888889,
       0.82222222, 0.86666667,
0.88888889, 0.86666667])
```

Calculate the accuracy of KNN for different values of k

KNN model evaluation

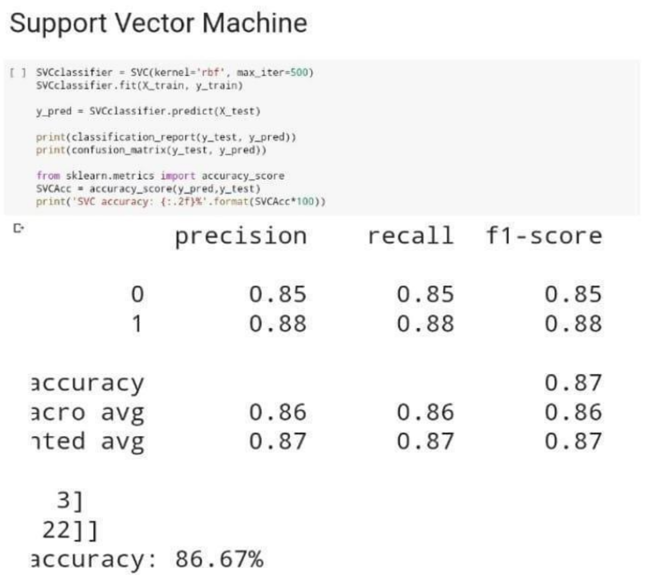
SUPPORT VECTOR MACHINE (SVM)

A common machine learning approach called Support Vector Machine (SVM) can be used for classification tasks like determining whether or not a bank loan is eligible. In order to maximize the margin between the hyperplane and the nearest data points, SVM seeks the best hyperplane for classifying the data into distinct groups.You would first need to gather data on previous loan applicants, including their demographics, financial history, credit score, and other pertinent criteria, in order to employ SVM for bank loan eligibility prediction. The SVM model would be trained using thisdata, and after being trained.Using a binary classification model with one class representing loan applicants who are eligible and the other class representing applicantswho are not eligible is one way to deploy SVM for bank loan eligibility prediction. With the input data as the features and the loan eligibility status as the labels, the SVM method would then be used tochoose the best hyperplane that divides these two classes.You might need to undertake feature engineering, preprocess the data,and modify the hyperparameters of the algorithm to increase the accuracy of the SVM model. Also, you might want to think about transforming the input data into a higher-dimensional space with a kernel function in order to better differentiate the various classes.Overall, SVM can be an effective tool for determining whether a person will qualify for a bank loan, but in order to make sure the model is accurate and reliable, it is crucial to carefully choose and preprocess the input data, tune the algorithm's hyperparameters, and assess the model's performance.

Formula for a Support Vector Machine (SVM) is as follows:

Although the prediction is based on the ideal hyperplane that is produced during the training process, the SVM algorithm does not have a precise formula for predicting bank loan eligibility.Nonetheless, the general procedures for employing SVM to forecastthe eligibility for bank loans can be summed up as follows:Gather information about previous loan applicants, such as theircontact details, financial background, credit rating, and other pertinent details. Create training and test sets from the data.Preprocessing the data entails handling missing values, normalizing or standardizing the features, and conducting any necessary feature engineering.Using the loan eligibility status as the goal variable and the otherattributes as the input variables, train the SVM model using the training data.Measure the model's performance using metrics like accuracy,precision, recall, and F1 score on the testing data.Predict the eligibility of new loan applicants based on their data usingthe trained SVM model.

Depending on the programming language and libraries used, the precise procedures for implementing SVM may change. The fundamental idea is still to use a machine learning algorithm to identify patterns in the data and then make predictions based on those patterns.


Support Vector Machine

```
[ ] SVCclassifier = SVC(kernel='rbf', max_iter=500)
    SVCclassifier.fit(X_train, y_train)

    y_pred = SVCclassifier.predict(X_test)

    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))

    from sklearn.metrics import accuracy_score
    SVCAcc = accuracy_score(y_pred,y_test)
    print('SVC accuracy: {:.2f}%'.format(SVCAcc*100))
```

```
              precision    recall  f1-score

           0       0.85      0.85      0.85
           1       0.88      0.88      0.88

 accuracy                              0.87
 acro avg          0.86      0.86      0.86
 nted avg          0.87      0.87      0.87

   3]
  22]]
 accuracy: 86.67%
```

SVM model evaluation

DECISION TREE (DT) ALGORITHM

Another well-liked machine learning approach that can be used to forecast bank loan eligibility is the Decision Tree (DT) algorithm. The decision tree algorithm constructs a model of decisions and potential outcomes that resembles a tree with the intention of determining the optimum path through the tree to forecast the target variable.You would first need to gather data on previous loan applicants,including their demographic details, financial history, credit score, and other pertinent characteristics, in order to employ thedecision tree algorithm for bank loan eligibility prediction. The decision tree model would be trained using this information, and it would then be able to forecast the eligibility of new loan applicants based on the information they provided.Gather information about previous loan applicants, such as their contact details, financial background, credit rating, and other pertinent details.Create training and test sets from the data.The following are the procedures for putting a decision treealgorithm for predicting bank loan eligibility into practice:Preprocessing the data entails handling missing values, normalizing or standardizing the features, and conducting any necessary feature engineering.Using the loan eligibility status as the goal variable and the other features as the input variables, train the decision tree model using the training data.Measure the model's performance using metrics like accuracy,precision, recall, and F1 score on the testing data.Recursively dividing the data into smaller subgroups according to thevalues of the input features, the decision tree algorithm then bases its decision on the subset with the target variable values that are the mosthomogeneous. Until a stopping requirement is satisfied, such as a maximum depth or a minimum amount of samples per leaf node, the splitting process continues.samples per leaf node, to increase the decision tree model's accuracy.Also, to merge many decision trees and enhance the predictive accuracy, you might want to think about employing an ensemble method like random forests.Overall, the decision tree algorithm can be a helpful tool for determining whether a person will qualify for a bank loan, but in order to ensure that the model is accurate and trustworthy, it is crucialto carefully choose and preprocess the input data, tune the algorithm'shyperparameters, and evaluate the model's performance.

Formula for the Decision Tree (DT) Algorithm is as follows:

Because the prediction is based on the tree-like model of actions and consequences that is developed during the training phase, the decision tree algorithm does not have a precise formula for predicting bank loan eligibility. Nonetheless, the general procedures for determining if a bank loan applicant is eligible can be summed up as follows:Gather information about previous loan applicants, such as their contact details, financial background, credit rating, and other pertinent details.Create training and test sets from the data.Preprocessing the data entails handling missing values, normalisingor standardising the features, and conducting any necessary featureengineering.Using the loan eligibility status as the goal variable and the other features as the input variables, train the decision tree model using the training data.Measure the model's performance using metrics like accuracy, precision, recall, and F1 score on the testing data.Predict the eligibility of new loan applicants based on their data usingthe trained decision tree model.Recursively dividing the data into smaller subgroups according to thevalues of the input features, the decision tree algorithm then bases itsdecision on the subset with the target variabledecision tree algorithm does not have a precise formula for predictingbank loan eligibility. Nonetheless, the general procedures for determining if a bank loan applicant is eligible can be summed up as follows:

Gather information about previous loan applicants, such as their contact details, financial background, credit rating, and other pertinent details.

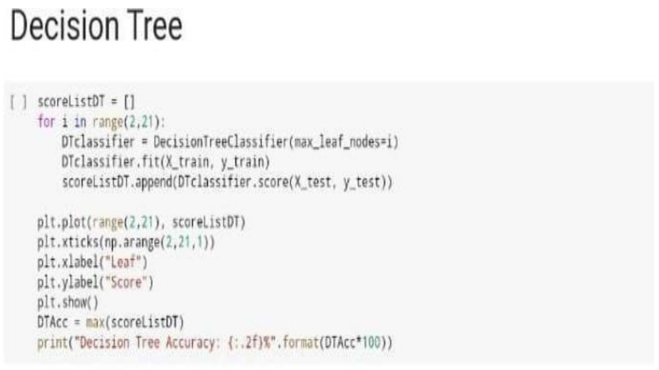
Create training and test sets from the data. Values that are the most homogeneous. Until a stopping requirement is satisfied, such as a maximum depth or a minimum amount of samples per leaf node, the splitting process continues.

The particular procedures for constructing decision tree algorithm may vary based on the programming language and libraries utilised. The fundamental idea is still to use a machine learning algorithm to identify patterns in the data and then make predictions based on those patterns.

You might need to adjust the hyperparameters of the algorithm, such as the maximum depth of the tree or the minimum number of samples per leaf node, to increase the decision tree model's accuracy.

Also, to merge many decision trees and enhance the predictive accuracy, you might want to think about employing an ensemble method like random forests.

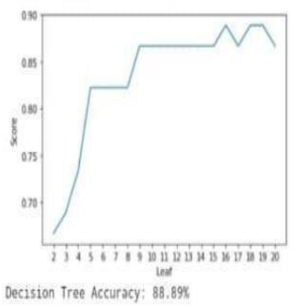
Overall, decision tree algorithm can be a useful tool for predicting bankloan eligibility, but it is important to carefully select and preprocess theinput data, tune the hyperparameters of the algorithm, and evaluate the model's performance to ensure that it is accurate and reliable.



Decision Tree

```
[ ] scoreListDT = []
    for i in range(2,21):
        DTclassifier = DecisionTreeClassifier(max_leaf_nodes=i)
        DTclassifier.fit(X_train, y_train)
        scoreListDT.append(DTclassifier.score(X_test, y_test))

    plt.plot(range(2,21), scoreListDT)
    plt.xticks(np.arange(2,21,1))
    plt.xlabel("Leaf")
    plt.ylabel("Score")
    plt.show()
    DTAcc = max(scoreListDT)
    print("Decision Tree Accuracy: {:.2f}%".format(DTAcc*100))
```



```
Decision Tree Accuracy: 88.89%
```

DT model evaluation

RANDOM FOREST (RF) MODEL

Indeed, I can assist you with it! An outline of how to forecast bank loan eligibility using an RF (Random Forest) model is given below:

Data Gathering: The first step is to gather the model's data. Past loan applicants' data will be included in this, including their income, credit score, employment status, loan amount, and whether they received a loan approval or denial.

Data preprocessing: After obtaining the data, this step is necessary. These tasks involve scaling the features, handling categorical variables, and deleting missing data.

Splitting the data into training and testing sets is the following step. The RF model will be trained using the training set, and its performance will be assessed using the testing set.

Model Training: With the training set, you can now train the RF model. The RF algorithm builds a group of decision trees, each of which is trained using a different random subset of the characteristics and data. The combined forecasts of each individual tree make up the RF model's final forecast.

Model Evaluation: After the RF model has been trained, you may assess how well it performed on the testing set. Accuracy, precision, recall, and F1 score are typical evaluation criteria for categorization issues like these.

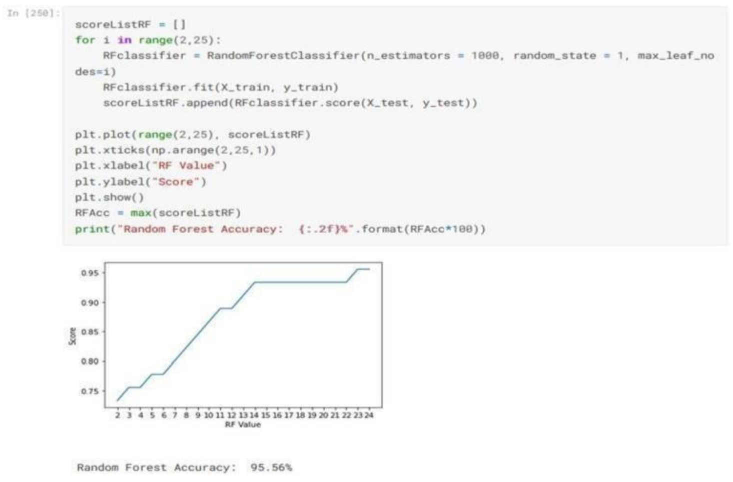
Lastly, you may utilise the trained RF model to predict outcomes based on fresh data. A new loan applicant's eligibility for a loan, for instance, may be predicted by entering certain information.

It's important to note that there are numerous machine learning algorithms that you might apply for this challenge in addition to RF models. Logistic regression, support vector machines, and neural networks are further common approaches for categorization issues. The most effective algorithm will vary depending on the size and complexity of your dataset, the available computing power, and the level of precision you require.

The method for calculating loan eligibility using an RF model will vary depending on how it is implemented. Nonetheless, the procedurefor forecasting with an RF model often entails the following steps:

The RF model receives a collection of input features for a loan application, such as their income, credit score, employment status, and loan amount.The probability of loan acceptance is then calculated by the modelusing a set of decision rules applied to the input features.The group of decision trees that make up the RF model are used to create the decision rules. Based on its own set of decision rules, eachdecision tree offers a "vote" for loan approval or refusal.The aggregate votes of every decision tree in the ensemble are then used to determine the RF model's final forecast.The outcome of the forecast can be continuous or binary (loanauthorised or denied) (probability of loan approval).

The criteria for approving a loan can be determined using the specifications of the particular lending organization.



```
In [250]:
scoreListRF = []
for i in range(2,25):
    RFclassifier = RandomForestClassifier(n_estimators = 1000, random_state = 1, max_leaf_no
des=i)
    RFclassifier.fit(X_train, y_train)
    scoreListRF.append(RFclassifier.score(X_test, y_test))

plt.plot(range(2,25), scoreListRF)
plt.xticks(np.arange(2,25,1))
plt.xlabel("RF Value")
plt.ylabel("Score")
plt.show()
RFAcc = max(scoreListRF)
print("Random Forest Accuracy:  {:.2f}%".format(RFAcc*100))
```

```
Random Forest Accuracy:  95.56%
```

RF model evaluation

GRADIENT BOOSTING MACHINE (GBM) MODEL

The general steps to estimate bank loan eligibility using gradient boosting machine learning are as follows:

Get the information required to train your machine learning model. Information on people who have previously sought for loans, such as their age, income, work status, credit score, and other pertinent details, should be included in this data.Data Preparation: You must clean and preprocess your data before you can train your machine learning model. In order to do this, missing values may be eliminated, categorical variablesmay be encoded, numerical features may be scaled, and other data cleaning techniques may be used.After your data has been cleansed and prepared, you can createnew features that could enhance the performance of your model. You could, for instance, add a new tool that calculates each person's debt to income ratio.Model selection: Following the preparation of your data, you must choose a machine learning model that is suitable for your issue. A gradient boosting machine model may be suitable for predicting bank loan eligibility since it can handle complex data and non-linear connections between features.Model Training: After choosing a model, you can train it withthe data you've collected. This entails optimising the model's performance by fitting the model to the training data and changing its parameters.Model Evaluation: You must assess your model's performance on a different test set after training it. By doing so, you'll be able to evaluate the model's accuracy and pinpoint any potentialimprovement areas.

Model Deployment: Lastly, you may use your trained model to project fresh applicants' loan eligibility. This can entail adding your model to a website or other piece of software.
Overall, thorough data preparation, feature engineering, model selection, and rigorous evaluation are required when creating a gradient boosting machine learning model for predicting bank loan eligibility.

The following is the formula for a gradient booster:

You can use these generic procedures to develop a Gradient Boosting Machine (GBM) model to forecast eligibility for bank loans: Preparation of Data: assemble and prepare the data. The data may need to be cleaned, missing values handled, and features transformed. Feature engineering is the process of developing additional features that can help the model perform better.Divide the data into training, validation, and test sets using the train-validation-test method.Pick a GBM algorithm, then adjust its hyperparameters using thevalidation set.Model training: Apply the chosen hyperparameters to the training set to train the GBM model.Model Evaluation: Assess the trained model's performance on the testdata.

An illustration formula for determining bank loan eligibility using GBM is shown below:
y = (1 / exp(-F(x))) where:

The expected likelihood that a loan would be approved is -y (ranging from 0 to 1)
F(x) is the weighted average of the forecasts from each individual decision tree:

F(x) = (wi * fi(x)), where I = 1 to n trees and -fi(x) is the prediction made by the i-th decision tree based on the input data x and -wi is the weight given to the i-th decision tree (determined during training)

Use a GBM library to implement this algorithm, such as XGBoost, Lite GBM, or Cat Boost, and adjust its hyperparameters to get the best results on the validation set. These are some examples of hyperparameters you might think about tuning:

learning rate: The reduction of the step size applied to avoid overfitting. Although more trees must be added to the model, a lower learning rate typically produces a better model.

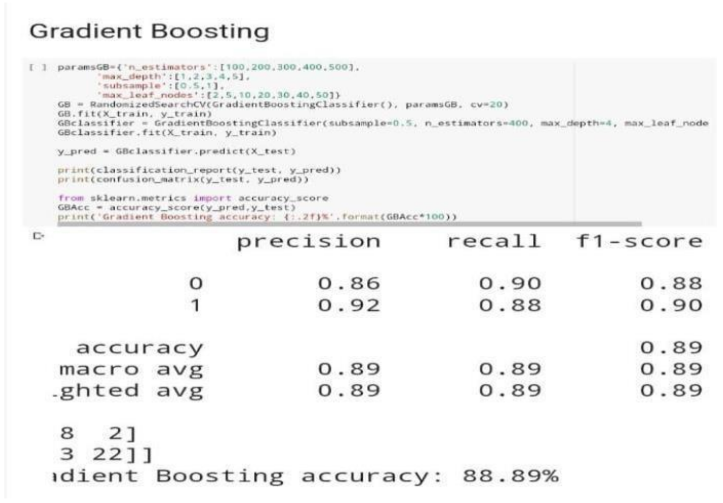
The number of trees in the GBM model is called n estimators. Although adding more trees can boost model accuracy, it can also lengthen training.

max depth: Each decision tree's maximum depth. Although a deeper tree may be able to capture more intricate links in the data, it risked overfitting the training set.

subsample: The percentage of samples that were used to create each distinct tree. A smaller subsample may help with generalization and avoid overfitting, but it may also produce a less accurate model.

Col sample by Tree: The percentage of features utilized by each distinct tree. A smaller feature subset can enhance generalization and reduce overfitting, but it may also lead to a less precise model.

Be aware that based on the particular problem and dataset you are working with, the precise formula and hyperparameters may change.



Gradient Boosting

```
[ ] paramsGB={'n_estimators':[100,200,300,400,500],
           'max_depth':[1,2,3,4,5],
           'subsample':[0.5,1],
           'max_leaf_nodes':[2,5,10,20,30,40,50]}
    GB = RandomizedSearchCV(GradientBoostingClassifier(), paramsGB, cv=20)
    GB.fit(X_train, y_train)
    GBclassifier = GradientBoostingClassifier(subsample=0.5, n_estimators=400, max_depth=4, max_leaf_node
    GBclassifier.fit(X_train, y_train)

    y_pred = GBclassifier.predict(X_test)

    print(classification_report(y_test, y_pred))
    print(confusion_matrix(y_test, y_pred))

    from sklearn.metrics import accuracy_score
    GBAcc = accuracy_score(y_pred,y_test)
    print('Gradient Boosting accuracy: {:.2f}%'.format(GBAcc*100))
```

```
              precision    recall  f1-score

           0       0.86      0.90      0.88
           1       0.92      0.88      0.90

    accuracy                           0.89
   macro avg       0.89      0.89      0.89
.ghted avg       0.89      0.89      0.89

 8  2]
 3 22]]
ıdient Boosting accuracy: 88.89%
```

GBM model evaluation

DISCUSSION

Predicting bank loan eligibility using machine learning is a popular and important application of data science in the financial industry. By using machine learning algorithms to analyze historical data, banks can develop predictive models that help them make informed decisions about whether or not toapprove loan applications.

The Gradient Boosting Machine (GBM), a form of ensemble learning approach that integrates the predictions of various decision trees, is one of the most frequently used machine learning algorithms for this purpose. GBM models can be used to represent intricate correlations between input features and loan eligibility because of their great accuracy and adaptability.

When utilising machine learning to forecast loan eligibility, there are a number of issues that must be resolved. Dealing with datasets that are imbalanced, where the proportion of approved loans to total loans is significantly larger than that of refused loans, is one of the largest issues. This can result in models that are biassed in favour of loan approvals, hence it calls for particular handling methods like resampling, weighting, or altering the decision threshold.

Making sure the model is impartial and fair to various demographic groups is another difficulty. Loan approval must be determined by objective standards, not by traits like racial, gender, or ethnic background, such as credit history, income, and debt-to-income ratio. To do this, the model must be verified for fairness using the proper metrics and approaches, and the input characteristics must be carefully chosen and assessed to make sure they are not linked with protected variables.

Finally, it's critical to make sure the model is clear and explicable so that regulators and loan applicants can understand how judgements are made. This necessitates the employment of interpretable models, such as decision trees or linear models, and the provision of detailed justifications for the usage of the input features in the predictions.

In general, forecasting bank loan eligibility with machine learning is a difficult task that calls for careful consideration of a variety of aspects, including data quality, feature engineering, algorithm selection, fairness, and transparency. Yet, banks can enhance their loan approval process and offer better services to their clients by utilising cutting-edge machine learning techniques.

CONCLUSION

In conclusion, applying machine learning to predict bank loan eligibility is a useful data science application that can aid banks in making defensible loan approval choices. Gradient Boosting Machines (GBMs), for example, have shown to be efficient at modelling intricate connections between input variables and loan eligibility and can offer high accuracy and flexibility.

However, predicting loan eligibility using machine learning also posesseveral challenges, including imbalanced datasets, fairness, and transparency. It is important to address these challenges by carefully selecting input features, using appropriate algorithms, and testing for fairness and transparency.

Overall, banks may enhance their loan approval process and offer better services to their clients by employing machine learning to anticipate loan eligibility. Banks can create strong and dependable models that are advantageous to the organisation and the loan applicants by paying close attention to the machine learning issues and best practises.

ADDITIONAL INFORMATION

The datasets examined and thorough documentation of the data analysis and model construction procedures can be found at: https://www.kaggle.com/orjiugochukwu/using-ml-algorithms-for-loan-approval-prediction.

REFERENCES

Here are some references and links that may be useful for predicting bank loan eligibility using machine learning:

a. "Loan Prediction Dataset" fromAnalytics Vidhya. This datasetcontains information about loan applicants and whether their loanwas approved or not, and can beused to train machine learning models for loan eligibilityprediction. https://datahack.analyticsvidhya.com/contest/practice-problem- loan-prediction-iii/

b. "Machine Learning for Credit Scoring" by Andreas Hoegger. This article discusses the use of machine learning for credit scoring and provides an overview of some of the most commonly used techniques. https://www.toptal.com/machine-learning/credit-scoring-machine- learning

c. "Credit Scoring and Loan DefaultPrediction using MachineLearning" by Yen-Ching Chang and Chia-Yen Lee. This researchpaper provides a comprehensivereview of the literature on credit scoring and loan default predictionusing machine learningtechniques. https://www.researchgate.net/publication/32574217_Credit_Scoring_and_Loan_Default_Prediction_Using_Machine_Learning

d. "Loan Eligibility Prediction usingMachine Learning Techniques" byKarthik Reddy and G. Ravi Kumar. This research paper discusses the use of machine learning techniques for loan eligibility prediction and compares the performance of various algorithms.https://www.ijraset.com/fileserve.php?FID=19468

e. "Machine Learning for LoanDefault Prediction: A Review" byXiaoming Li and Xiaohua Hu.This article provides acomprehensive review of theliterature on loan defaultprediction using machine learningtechniques. https://www.mdpi.com/2076- 3417/9/7/1469/htm

f. "Loan Prediction Dataset" fromAnalytics Vidhya. This datasetcontains information about loan applicants and whether their loanwas approved or not, and can beused to train machine learning models for loan eligibilityprediction. https://datahack.analyticsvidhya.com/contest/practice-problem- loan-prediction-iii/

g. "Machine Learning for Credit Scoring" by Andreas Hoegger. This article discusses the use of machine learning for credit scoring and provides an overview of some of the most commonly used techniques. https://www.toptal.com/machine-learning/credit-scoring-machine- learning

h. "Credit Scoring and Loan DefaultPrediction MachineLearning" by Yen-Ching Changand Chia-Yen Lee. This researchpaper provides a comprehensivereview of the literature on creditscoring and loan default predictionusing machine learning techniques. https://www.researchgate.net/publication/325742177_Crdt_Scoring_and_Loan_Default_Prediction_Using_Machine_Learning

i. "Loan Eligibility Prediction using Machine Learning Techniques" byKarthik Reddy and G. Ravi Kumar. This research paper discusses the use of machine learning techniques for loan eligibility prediction and compares the performance of various algorithms. https://www.ijraset.com/fileserve.php?FID=19468

j. "Machine Learning for LoanDefault Prediction: A Review" byXiaoming Li and Xiaohua Hu.This article provides acomprehensive review of the literature on loan defaultprediction using machine learningtechniques. https://www.mdpi.com/2076-3417/9/7/1469/html

k. "Loan Prediction Dataset" fromAnalytics Vidhya. This datasetcontains information about loanapplicants and whether their loanwas approved or not, and can beused to train machine learningmodels for loan eligibilityprediction. https://datahack.analyticsvidhya.com/contest/practice-problem- loan-prediction-iii/

l. "Machine Learning for Credit Scoring" by Andreas Hoegger. This article discusses the use of machine learning for credit scoring and provides an overviewof some of the most commonlyused techniques. https://www.toptal.com/machine-learning/credit-scoring-machine- learning

m. "Credit Scoring and Loan DefaultPrediction using MachineLearning" by Yen-Ching Changand Chia-Yen Lee. This researchpaper provides a comprehensivereview of the literature on creditscoring and loan default predictionusing machine learningtechniques. https://www.researchgate.net/publication/325742177_Credit_Scoring_and_Loan_Default_Prediction_Using_Machine_Learning

n. "Loan Eligibility Prediction using Machine Learning Techniques" byKarthik Reddy and G. Ravi Kumar. This research paper discusses the use of machine learning techniques for loan eligibility prediction and compares the performance of various algorithms. https://www.ijraset.com/fileserve.php?FID=19468

o. "Machine Learning for LoanDefault Prediction: A Review" byXiaoming Li and Xiaohua Hu.This article provides acomprehensive review of the literature on loan defaultprediction using machine learningtechniques. https://www.mdpi.com/2076-3417/9/7/1469/html

DB14

ORIGINALITY REPORT

10%	5%	6%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Ugochukwu. E. Orji, Chikodili. H. Ugwuishiwu, Joseph. C. N. Nguemaleu, Peace. N. Ugwuanyi. "Machine Learning Models for Predicting Bank Loan Eligibility", 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), 2022 Publication	2%
2	Submitted to University of North Texas Student Paper	1%
3	www.researchgate.net Internet Source	1%
4	B.Pitchai Manickam, P Kamalakannan, S Jayakumar, S Padmanaban, M Nithish Balaji. "Analysis on Forecasting of Loan using Supervised Machine Learning Models", 2022 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI), 2022 Publication	1%
5	Submitted to University of Sunderland Student Paper	1%

6	J. Y.-H. Pong. "A comparative study of two automatic document classification methods in a library setting", Journal of Information Science, 07/12/2007 Publication	1%
7	esling.github.io Internet Source	<1%
8	S. Varshaa Sai Sripriya, Sai Divya Santoshi Varrey, M Venkateshkumar. "Predictive Model to Compute Eligibility Test for Loans", 2022 IEEE Industrial Electronics and Applications Conference (IEACon), 2022 Publication	<1%
9	Paola Campadelli, Elena Casiraghi, Diana Artioli. "A Fully Automated Method for Lung Nodule Detection From Postero-Anterior Chest Radiographs", IEEE Transactions on Medical Imaging, 2006 Publication	<1%
10	marunouchiofficegroups.com Internet Source	<1%
11	www.semanticscholar.org Internet Source	<1%
12	emeritus.org Internet Source	<1%
13	towardsdatascience.com Internet Source	<1%

14	Submitted to Bournemouth University Student Paper	<1%
15	Submitted to University of Hertfordshire Student Paper	<1%
16	Submitted to Singapore Institute of Technology Student Paper	<1%
17	bmcmedinformdecismak.biomedcentral.com Internet Source	<1%
18	Jozef Legeny, Raquel Viciana-Abad, Anatole Lecuyer. "Toward Contextual SSVEP-Based BCI Controller: Smart Activation of Stimuli and Control Weighting", IEEE Transactions on Computational Intelligence and AI in Games, 2013 Publication	<1%
19	Submitted to The University of Memphis Student Paper	<1%
20	Submitted to University of Hong Kong Student Paper	<1%
21	faircentp2p.weebly.com Internet Source	<1%
22	Submitted to La Trobe University Student Paper	<1%
23	www.extrica.com	

Internet Source

<1%

---

24 www.qeios.com
Internet Source
<1%

---

25 H. Kargupta, H. Dutta. "Orthogonal Decision Trees", Fourth IEEE International Conference on Data Mining (ICDM'04), 2004
Publication
<1%

---

26 Hossein Moayedi, Atefeh Ahmadi Dehrashid. "A new combined approach of two neural-metaheuristic techniques based on the Cuckoo optimization algorithm and backtracking search algorithms for predicting and appraisal of landslide susceptibility mapping", Research Square Platform LLC, 2023
Publication
<1%

---

27 diva-portal.org
Internet Source
<1%

---

28 www.asdl.gatech.edu
Internet Source
<1%

---

29 www.dgtl-factory.com
Internet Source
<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On